



Usando la taglib core

Transcripción

[00:00] Hola a todos y todas, bienvenidos y bienvenidas una vez más a este curso de Servlet. Vimos en el video pasado que nosotros habíamos colocado dentro de nuestro JSP de listar empresas, colocamos JSTL junto con Expression Language para mejorar todo lo que es el mantenimiento de los archivos JSP, ya que colocar scriplets no es una buena idea porque el mantenimiento realmente se vuelve un infierno en el futuro si tenemos mucho código ahí.

[00:30] Entonces ese JSTL más expression language nos ayuda bastante a mejorar la claridad de lo que estamos leyendo en nuestro documento, en nuestros archivos. Entonces vamos a aprovechar que tenemos JSTL y vamos a ver algunas otras tecnologías que tenemos dentro JSTL.

[00:49] Ya vemos que tenemos nuestro core, que sirve para el control de flujos, y además de él tenemos fmt que sirve para hacer formatación de por ejemplo fechas y números y i18n, que es internacionalización, que nos va a ayudar a que, por ejemplo, nosotros creamos una página en español y después la traducción a inglés, francés, alemán, etcétera. Es mucho más fácil.

[01:13] Además de esas dos tecnologías, nosotros tenemos la parte de SQL que no es muy recomendado utilizar SQL con JSTL ya que no tiene mucho sentido hacer consultas a nuestra base de datos dentro de nuestro JSP. No tiene sentido, no lo utilicen. Y además es posible generar XML, que yo en los proyectos en los que he estado no los he visto realmente, no he visto utilizar esa parte de la biblioteca.

[01:45] Sin embargo está ahí, pero nosotros vamos a hacer foco en core y fmt. Para poder utilizar core dentro de nuestro JSP tenemos que colocar esa taglib con la uri, `java.sun.com/jsp/jstl/core`. Y para fmt lo mismo, solo que terminando en fmt en vez de core.

[02:09] Y a su vez, nosotros tenemos que colocar un prefijo que sirve para que nosotros podamos leer esa biblioteca y utilizar esa biblioteca dentro de nuestras tags, dentro de nuestro JSP. Entonces, en el caso de core, normalmente se utiliza el prefijo “c”, y en el caso de fmt, se utiliza normalmente el prefijo fmt.

[02:28] Pero claro, ustedes lo pueden cambiar con el nombre que ustedes quieran a ese “c” y ese “fmt”. Perfecto, vamos a ir a nuestros archivos y ahora lo que yo quiero es hacer un cambio dentro de ese `formNuevaEmpresa` ya que nosotros lo tenemos como un HTML y hay algunas cosas que me gustaría cambiar ahí.

[02:46] Entonces para eso vamos a hacer un clic en el archivo `F2` y vamos a cambiarlo a JSP para poder aprovechar nuestro JSTL. Bien, ya una vez cambiado voy a hacer un “Ctrl + Shift + W”, cerramos todas las pestañitas que tenemos acá. Entonces voy a hacer doble clic en ese `formNuevaEmpresa` y acá lo que voy a hacer es traer nuestro core que teníamos ya en nuestra lista de empresas `.jsp`.

[03:17] Doble clic en `lista de empresas.jsp` y voy a copiar todo este taglib con “Ctrl + C”, voy a ir acá arriba y vamos a hacer un “Ctrl + V”. “Ctrl + S” para guardar y si a ustedes se les ve en negro esta parte significa que no ha reconocido todavía la librería, entonces para eso pueden cerrar el archivo y abrirlo de nuevo ese form.

[03:44] El Eclipse IDE no es muy bueno con el tratamiento de los archivos `.jsp`, sí es bueno con la parte de Servlet y archivos Java, pero con JSP aún tiene algunas cosas para arreglar. Bien, nosotros tenemos nuestro core ¿y qué es lo

que vamos a utilizar? ¿Qué es lo que vamos a hacer? Vean que dentro de nuestro form action, nosotros tenemos un camino acá que ese camino va a ir para nuevaEmpresa.

[04:11] Pero además de nuevaEmpresa, nosotros hemos colocado acá gerenciador, que es el nombre, el contexto que nosotros tenemos dentro de nuestro proyecto. ¿Qué es ese contexto? Bueno, déjenme hacer lo siguiente. Esto que voy a hacer ahora, no es necesario que ustedes lo hagan en sus casas, yo es solo para mostrarles lo que ocurre.

[04:49] Voy a eliminar este gerenciador, nuestro proyecto de nuestro Tomcat, solamente lo desvinculé, y ahora nos vamos a ir a nuestro gerenciador, clic derecho en propiedades y acá abajo, dentro de web project settings, dejen que aumente un poco el tamaño acá, web Project settings, nosotros tenemos esto. ¿Qué es ese contexto?

[04:50] Ese contexto es la parte de la URL luego de nuestro localhost: 8080, después de la puerta, lo que sigue después es un contexto, después de nuestra IP viene el contexto, ese contexto es la forma en que nuestro Tomcat tiene para reconocer que vamos a llamar a nuestro proyecto, en este caso gerenciador. Sin embargo, este contexto nosotros lo podemos cambiar por el nombre que quisiéramos.

[05:20] Por ejemplo puedo poner contexto X. Voy a dar un apply, y voy a enviar de nuevo nuestro proyecto a nuestro Tomcat. Ahí está dentro de nuestro Tomcat, voy a dar un run y vemos que en la consola está todo bien, vamos a ver dentro de nuestro Google Chrome, nosotros tenemos nuestro gerenciador/listaEmpresas, si doy un enter ahora nos está diciendo: “no encontré esa página”.

[05:48] Sin embargo, si yo pongo acá gerenciadorX, doy enter, ahora sí encontró la página. Entonces vean que ese contexto puede mudar, puede cambiar, en el futuro es posible cambiar eso por cualquier otro nombre,

entonces no es una buena idea tener hardcodeado el nombre del contexto acá dentro de un formulario dentro de nuestro JSP.

[06:14] Entonces, ¿cómo vamos a hacer para cambiar eso, para eliminar ese contexto? Bueno, lo que nosotros tenemos que hacer es utilizar justamente nuestro core. ¿Qué vamos a hacer con nuestro cuerpo? Vamos a ver lo siguiente, voy a abrir una tag acá C: para usar nuestro core y vamos a usar url.

[06:34] Enter, no es necesario esta parte acá, ya directamente lo voy a cerrar en este momento. Nosotros con un “Ctrl + espacio” podemos ver ese value. Ese value es el nombre de lo que viene luego de nuestro contexto. O sea, nosotros vamos a poder agregar alguna cosa a nuestro contexto, “/nuevaEmpresa” que sería esto de acá es lo mismo que esto de acá, solo que con nuestro contexto, cambiando dependiendo cuál sea realmente.

[07:09] Ahora en este momento no tenemos gerenciarX, entonces nos colocaría gerenciadorX/nuevaEmpresa. Si le colocamos otro nombre, cambiaría también otro nombre /nuevaEmpresa. Entonces vamos a hacer lo siguiente. Voy a mostrar qué es lo que vamos a ver acá.

[07:26] Nuestro server va a reiniciarse. No era necesario reiniciar, ya que es un JSP. Bueno, voy a dar un clic en enter, gerenciadorX/nuevaEmpresa. ¿Cómo era el nombre? Dentro de nuestro formNuevaEmpresa. Entonces voy a ir a formNuevaEmpresa.jsp y vean, acá tenemos nuestro contexto que ha cambiado.

[07:53] Entonces consigue agarrar ese contexto, vamos a hacer lo siguiente. En vez de colocar /gerenciador/nuevaEmpresa, vamos a colocar directamente esto de acá. Esto que estoy haciendo no es muy lindo, sin embargo funciona. Entonces voy a hacer un "Ctrl + S", vamos a hacer un run de nuevo, y si nosotros nos vamos al código, vean que ahora ha cambiado ese action cambió a nuestro contexto, que es lo que les había dicho.

[08:21] Sin embargo, ahora yo no quiero usar ese gerenciadorX, entonces vamos a ir a nuestro proyecto, voy a parar mi máquina virtual. Todo esto que les mostré no es necesario que ustedes lo hagan, voy a eliminar nuestro gerenciador de acá, deslinkear. Vamos a hacer clic derecho en nuestro gerenciador propiedades, dentro de nuestro project settings voy a volver a gerenciador. Aplicar cambios, voy a colocarlo de nuevo dentro de nuestro Tomcat.

[08:56] Voy a dar un run. Bien. Y ahora solo para estar seguro voy a dar un F5 ya no existe más nuestro gerenciadorX, ahora existe gerenciador. Vamos a dar un enter y si veo el código fuente está gerenciador, cambió de nuevo. Pero como les dije, esto de acá no está muy bien, no se ve muy bien. ¿Entonces, qué es lo que vamos a hacer?

[09:18] Vamos a sacar esto de acá, vamos a subirlo y vamos a hacer lo siguiente, Vamos a hacer un “Ctrl + espacio” dentro de nuestro URL y vamos a colocar un var, vamos a crear, vamos a hacer que ese valor que obtenemos a través de nuestro URL value, lo podamos acceder a través de una variable.

[09:42] ¿Cuál va a ser el nombre de esa variable? Va a ser “LinkServletNuevaEmpresa”. “Ctrl + S”, entonces, ahora tenemos ese todo ese contexto y /nuevaEmpresa dentro de una variable. ¿Cómo hacemos para usar eso acá? Ya sabemos que es a través de expression language, entonces signo de dólar y abrimos llaves y vamos a colocar simplemente “\${LinkServletNuevaEmpresa}”.

[10:09] “Ctrl + S” y con esto ya debería funcionar nuestro form. Bien, voy a hacer un F5. Vamos a ver nuestra página y perfecto, está funcionando, si yo acá pongo Alura, envío, vemos nuestro Alura dentro de nuestra nuevaEmpresa, pero ahora que estamos dentro de nuestra nueva empresa ha surgido un nuevo problema. ¿Qué ocurre si yo simplemente llamo a nuestro Servlet nuevaEmpresa?

[10:35] Voy a dar un enter y nos está diciendo Method not allowed. En vez de hacer eso, nosotros vamos a llamar a nuestra nueva empresa registrada, perdón, había llamado nuevaEmpresa, no era eso lo que quería hacer. Yo quería hacer nuevaEmpresaRegistrada.jsp.

[10:56] ¿Qué es lo que ocurre si yo llamo a esa nuevaEmpresaRegistrada.jsp? Vean que dice empresa registrada y no tiene ningún nombre, eso realmente no está bien, o sea, no hemos registrado ninguna empresa, entonces lo que tenemos que hacer, lo que quiero hacer es cambiar ese mensaje. ¿Cómo hacemos para cambiar ese mensaje de nuevo JSTL?

[11:16] Dependiendo de si nosotros recibimos una empresa o no, yo quiero que ese mensaje cambie, entonces vamos a hacer de nuevo agarrar ese taglib "Ctrl + C", vamos a hacer un "Ctrl + V" acá, "Ctrl + S" y acá, nosotros ya podemos hacer lo siguiente, nosotros podemos colocar un condicional, un if. ¿Cómo? Abriendo tag c: if.

[11:41] Vamos a dar un enter. Voy a dar un espacio acá y colocar esto acá adentro. Acá bien. Y vean lo siguiente: nosotros dentro de ese if tenemos un test. Dentro de ese test es donde nosotros vamos a colocar la lógica para comprender si tenemos que ejecutar una cosa o no. Ese nombre test viene de esta Taglib, obtuvo varios nombres de funciones de un xml anterior, entonces, por eso heredó ese nombre de test.

[12:14] Sé que es un poco difícil, pero simplemente no hay mucho para aprender realmente de lo que es esa sintaxis de JSTL. Entonces en este caso, nosotros vamos a usar expression languages de nuevo colocando un signo de dólar, abrimos llave y acá adentro lo que nosotros queremos hacer es preguntar, si no es vacío nuestra empresa, si lo que tenemos dentro de empresa no es vacía, entonces muestra el mensaje empresa.

[12:46] La primera empresa registrada. Y ahora, en caso de que la empresa esté vacía realmente o sea, vamos a sacar ese not, yo quiero mostrar otro mensaje.

¿Cuál mensaje? Ninguna empresa registrada. Esos van a hacer los dos mensajes que vamos a poder mostrar. "Ctrl + S", vamos a reiniciar nuestro servidor y vamos a ver qué pasa.

[13:11] Si yo llamo directamente a nuestro JSP vamos a ver que ninguna empresa registrada. Perfecto, y ahora quiero ver el caso contrario, el caso de que tengamos una empresa, voy a ir a nuestro form. No es HTML que es JSP. Y acá voy a colocar Google, vamos a enviar y ahora efectivamente funciona. Empresa Google registrada!

[13:33] Entonces vean que nosotros conseguimos utilizar nuestra librería, nuestra parte core de la librería de JSTL, de otras formas para poder mejorar nuestra NuevaEmpresaRegistrada y nuestro form de nuevaEmpresa. Bien, ahora les dije que les iba a enseñar sobre fmt y así va a ser. Así que hagan los ejercicios y nos vemos en el próximo video.